**Project Description:**

Note: This is one of the two complementary competitions that together comprise the M5 forecasting challenge. Can you estimate, as precisely as possible, the point forecasts of the unit sales of various products sold in the USA by Walmart? If you are interested in estimating the uncertainty distribution of the realized values of the same series, be sure to check out its [*companion competition*](https://www.kaggle.com/c/m5-forecasting-uncertainty)

How much camping gear will one store sell each month in a year? To the uninitiated, calculating sales at this level may seem as difficult as predicting the weather. Both types of forecasting rely on science and historical data. While a wrong weather forecast may result in you carrying around an umbrella on a sunny day, inaccurate business forecasts could result in actual or opportunity losses. In this competition, in addition to traditional forecasting methods you’re also challenged to use machine learning to improve forecast accuracy.

The Makridakis Open Forecasting Center (MOFC) at the University of Nicosia conducts cutting-edge forecasting research and provides business forecast training. It helps companies achieve accurate predictions, estimate the levels of uncertainty, avoiding costly mistakes, and apply best forecasting practices. The MOFC is well known for its Makridakis Competitions, the first of which ran in the 1980s.

In this competition, the fifth iteration, you will use hierarchical sales data from Walmart, the world’s largest company by revenue, to forecast daily sales for the next 28 days. The data, covers stores in three US States (California, Texas, and Wisconsin) and includes item level, department, product categories, and store details. In addition, it has explanatory variables such as price, promotions, day of the week, and special events. Together, this robust dataset can be used to improve forecasting accuracy.

If successful, your work will continue to advance the theory and practice of forecasting. The methods used can be applied in various business areas, such as setting up appropriate inventory or service levels. Through its business support and training, the MOFC will help distribute the tools and knowledge so others can achieve more accurate and better calibrated forecasts, reduce waste and be able to appreciate uncertainty and its risk implications.

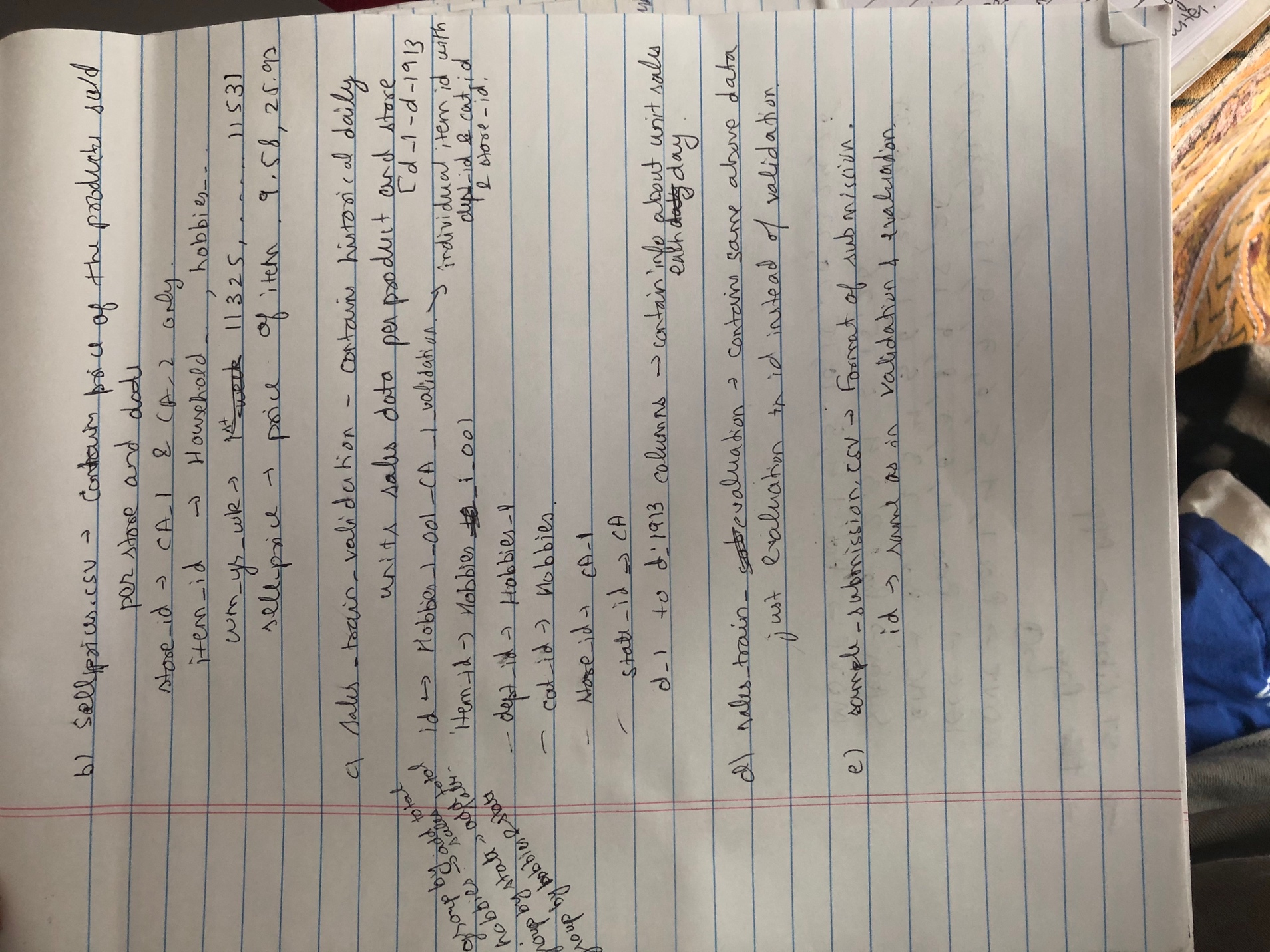
**Data:**

In the challenge, you are predicting item sales at stores in various locations for two 28-day time periods. Information about the data is found in the [M5 Participants Guide](https://mofc.unic.ac.cy/m5-competition/).

* calendar.csv - Contains information about the dates on which the products are sold.
* sales\_train\_validation.csv - Contains the historical daily unit sales data per product and store [d\_1 - d\_1913]
* sample\_submission.csv - The correct format for submissions. Reference the [Evaluation](https://www.kaggle.com/c/m5-forecasting-accuracy/overview/evaluation) tab for more info.
* sell\_prices.csv - Contains information about the price of the products sold per store and date.
* sales\_train\_evaluation.csv - Includes sales [d\_1 - d\_1941] (labels used for the Public leaderboard)

A close up of a piece of paper

Description automatically generated



**Python files Description:**

1. **forecast\_toper\_model\_kaggle-** This file is about the lgb method used. Error is 0.74. That is leaf based tree boosting method. Here we have created many other features as described below:

First off what each feature mathematically does.

1. lag\_7: sales shifted 7 steps downwards *for each group*. The example above focuses on one group only as an example. That is why the first value appears on the 7th index.
2. lag\_28: sales shifted 28 steps downwads. That is why the first value appears on the 28th index.
3. rmean\_7\_7: rolling mean sales of a window size of 7 over column lag\_7. First value (0.2857) appears on the 13th index because means including nan are nan.
4. rmean\_7\_28: rolling mean sales of a window size of 7 over column lag\_28. First value (0.357) appears on the 34th index because that is the first time the mean formula gets all 7 non-nan values.
5. rmean\_28\_7: rolling mean sales of a window size of 28 over column lag\_7. First value (0.2857) appears on the 3th index because it is the first time the mean formula gets 28 non-nan values.
6. rmean\_28\_28: rolling mean sales of a window size of 28 over column lag\_28. First value appears on 55th index because that is the first time the formula here all non-nan values.

The intuition as far as I can understand is the following:

* 1. Captures the week-on-week similarity and that too of just the past week. In other words, people are likely to shop this monday similar to the last monday (except it is some special occassion).
* 2. Captures the weekly similarity from a month-to-month perspective. Example: people in the 1st weekend of a month shop more so that weekend looks more similar to first weeks of other months than the previous weekend. (Though 28 is arguable here. A month is generally 30. Interesting would be a variable window depending on when the comparative week starts. Dealing with edge cases like week divided into 2 months will be tricky).

Since individual data points are prone to erratic spikes or troughs, mean provides a more "representative" picture.

* 3. Captures the information regarding the sales of the whole *previous week ending 7 days in the past* i.e. if we are at day 14, then the average is of sales from days 1-7 NOT days 7-14. This provides the information about the whole week and not just a single day sale comparison like lag\_7 to bring the lag\_7 value into "better weekly context".
* 4. Captures the information regarding the sales of the entire *previous 4 weeks ending 7 days in the past* i.e. if we are at day 35, then the average is sales from days 1-28.
* 5. Captures the information regarding the sales of the whole *week ending 4 weeks ago* i.e. if we are on day 35, then the average is of sales from day 1-7. (Assuming for simplicity the month is 28 days), this provides the information of not just a month-to-month comparison of the same day (day 7 of month one vs day 7 of month two), but the entire week leading up to day 7. Again the idea I believe is to capture the whole week and not just a single day sale comparison like lag\_28 to bring the lag\_28 value into "better weekly context".
* 6. Captures the information regarding the sales of the entire *previous 4 weeks ending 4 weeks in the past* i.e. if we are at day 56, then the average is of days 1-28. (Assuming for simplicity the month is 28 days), the idea again is to bring the point value of lag\_28 into a better context (i.e. of day 28 when being compared to day 56) into a "better monthly context".

How would you "talk" about these features?

* Hey let's see how the sales were last friday compared to this friday?
* Hey let's see how the sales were first weekend of the last month compared to first weekend of this month?
* May be comparing last saturday to this saturday is too specific. Week-on-week same day trends are more likely to be similar if the prior week went similar too. It would make sense to not just have the last saturday but also the mean of the whole week leading upto that day to give the model the "hint" how *normal* the whole week was.

1. divyank\_M5-forecasting-eda- Here prophet method is used without doing parameterization. Due to this error is 1.73 which is very large.
2. M5-forecasting-eda- This is just a to do some visualization based on store, dep, place. No model
3. Lstm- Here we used LSTM to do the forecasting. Error was 0.85. We use RNN generator. No such parameterization was done.